Theory Meets Practice
…it's about TIME!

Roger Wattenhofer
„People who are really serious about software should make their own hardware.”

Alan Kay
„People who are really serious about algorithms should make their own software.”
Angle-of-Arrival Measurements with SpiderBat

Receiver

Sender

South

North

East

West

Time [ms]
Science Fiction: Learning Environment with SpiderBat?!

[Sommer et al., IPSN 2011]
An Example for Theory?

• Fully connected network
• Between any pair of nodes one message/round capacity
• Task: \( n \) nodes need to send/receive up to \( n \) messages
• How many rounds does it take until all messages are received?
Instead?

- Information Theory: Routing $n^2$ msgs with total capacity $n^2$ takes $O(1)$ time!? 

- Theorem: No, rather $\Theta(\log^* n)$.

- **Distributed** Information Theory?

[Lenzen et al., STOC 2011]
Theory *Meets* Practice?
Practice: Sensor Networks
Example: Dozer

- Up to 10 years of network life-time
- Mean energy consumption: 0.066 mW
- Operational network since 2007
- High availability, reliability (99.999%)

[Burri et al., IPSN 2007]
Energy-Efficient Protocol Design

• Communication subsystem is the main energy consumer
  – Power down radio as much as possible

<table>
<thead>
<tr>
<th>TinyNode</th>
<th>Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>uC sleep, radio off</td>
<td>0.015 mW</td>
</tr>
<tr>
<td>Radio idle, RX, TX</td>
<td>30 – 40 mW</td>
</tr>
</tbody>
</table>

• Issue is tackled at various layers
  – MAC
  – Topology control / clustering
  – Routing

→ Orchestration of the whole network stack to achieve duty cycles of ~ 0.1%
Dozer in Action
Is Dozer a theory-meets-practice success story?

• **Good** news
  – Theory people can develop good systems!
  – Sensor network (systems) people write that Dozer is one of the “best sensor network systems papers”, or: “In some sense this is the first paper I'd give someone working on communication in sensor nets, since it nails down how to do it right.”

• **Bad** news: Dozer does not have an awful lot of theory inside

• **Ugly** news: Dozer v2 has even less theory than Dozer v1
no theory 😞
Theory for sensor networks, what is it good for?!

How many lines of pseudo code //
Can you implement on a sensor node?

The best algorithm is often complex //
And will not do what one expects.

Theory models made lots of progress //
Reality, however, they still don’t address.

My advice: invest your research £££s //
in ... impossibility results and lower bounds!
Example: Clock Synchronization
...it's about TIME!
Clock Synchronization
Clock Synchronization in Networks

- **Global Positioning System (GPS)**
- **Radio Clock Signal**
- **AC-power line radiation**
- **Synchronization messages**
Clock Synchronization in Networks

- **Global Positioning System (GPS)**
- **Radio Clock Signal**
- **AC-power line radiation**
- **Synchronization messages**

- Imagery and symbols related to each topic.
Problem: Physical Reality

clock rate

message delay

\[
1 + \epsilon \\
1 \\
1 - \epsilon
\]
Clock Synchronization in Networks?

- *Time, Clocks, and the Ordering of Events in a Distributed System*
- *Internet Time Synchronization: The Network Time Protocol (NTP)*
- *Reference Broadcast Synchronization (RBS)*
  J. Elson, L. Girod and D. Estrin, OSDI 2002
- *Timing-sync Protocol for Sensor Networks (TPSN)*
  S. Ganeriwal, R. Kumar and M. Srivastava, SenSys 2003
- *Flooding Time Synchronization Protocol (FTSP)*
  M. Maróti, B. Kusy, G. Simon and Á. Lédeczi, SenSys 2004
- and many more ...

FTSP: State of the art clock sync protocol for networks.
Variants of Clock Synchronization Algorithms

Tree-like Algorithms
e.g. FTSP

Distributed Algorithms
e.g. GTSP
[Sommer et al., IPSN 2009]

All nodes consistently average errors to all neighbors
FTSP vs. GTSP: Global Skew

- Network synchronization error (global skew)
  - Pair-wise synchronization error between any two nodes in the network

FTSP (avg: 7.7 μs)  
GTSP (avg: 14.0 μs)
FTSP vs. GTSP: Local Skew

- Neighbor Synchronization error (local skew)
  - Pair-wise synchronization error between neighboring nodes

FTSP (avg: 15.0 μs)  
GTSP (avg: 2.8 μs)
Time in (Sensor) Networks

- Sensing: Local
- TDMA: Local
- Localization: Global
- Duty-Cycling: Local

Clock Synchronization Protocol

Hardware Clock
Clock Synchronization in Theory?

- Given a communication network
  1. Each node equipped with hardware clock with drift
  2. Message delays with jitter

- Goal: Synchronize Clocks (“Logical Clocks”)
  - Both global and local synchronization!
Time Must Behave!

• Time (logical clocks) should **not** be allowed to **stand still** or **jump**

• Let’s be more careful (and ambitious):
  • Logical clocks should **always move forward**
    • Sometimes faster, sometimes slower is OK.
    • But there should be a minimum and a maximum speed.
    • As close to correct time as possible!
Local Skew

Tree-like Algorithms
e.g. FTSP

Distributed Algorithms
e.g. GTSP

Bad local skew
Synchronization Algorithms: An Example ("A^{max}")

- **Question:** How to update the logical clock based on the messages from the neighbors?

- **Idea:** Minimizing the skew to the fastest neighbor
  - Set clock to maximum clock value you know, forward new values immediately

- **First all messages are slow (1) ...**
  ... then suddenly all messages are fast (0)!

![Diagram showing synchronization concept](image-url)
Local Skew: Overview of Results

Everybody’s expectation, 10 years ago („solved“)

Lower bound of \( \log D / \log \log D \)
[Fan & Lynch, PODC 2004]

Tight lower bound
[Lenzen et al., PODC 2009]

Kappa algorithm
[Lenzen et al., FOCS 2008]

1
\( \log D \)
\( \sqrt{D} \)
\( D \)
...
Enforcing Clock Skew

- Messages between two neighboring nodes may be fast in one direction and slow in the other, or vice versa.
- A constant skew between neighbors may be „hidden“.
- In a path, the global skew may be in the order of $D/2$. 
Local Skew: Lower Bound

Theorem: $\Omega\left(\frac{\log(\beta-1)}{\epsilon} D\right)$ skew between neighbors

- Add $l_0/2$ skew in $l_0/(2\epsilon)$ time, messing with clock rates and messages
- Afterwards: Continue execution for $l_0/(4(\beta-1))$ time (all $h = 1$)
  - Skew reduces by at most $l_0/4$ → at least $l_0/4$ skew remains
  - Consider a subpath of length $l_1 = l_0 \cdot \epsilon/(2(\beta-1))$ with at least $l_1/4$ skew
  - Add $l_1/2$ skew in $l_1/(2\epsilon) = l_0/(4(\beta-1))$ time → at least $3/4 \cdot l_1$ skew in subpath
- Repeat this trick ($+\frac{1}{2}, -\frac{1}{4}, +\frac{1}{2}, -\frac{1}{4}, ...$) $\log_{2(\beta-1)/\epsilon} D$ times
Surprisingly, up to small constants, the $\Omega(\log_{(\beta-1)/\epsilon} D)$ lower bound can be matched with clock rates $\in [1, \beta]$ (tough part, not in this talk).

We get the following picture [Lenzen et al., PODC 2009]:

<table>
<thead>
<tr>
<th>max rate $\beta$</th>
<th>$1+\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>local skew</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

... because too large clock rates will amplify the clock drift $\epsilon$.

We can have both smooth and accurate clocks!
Local Skew: Upper Bound

- Surprisingly, up to small constants, the $\Omega(\log (\bar{\beta}/\epsilon) D)$ lower bound can be matched with clock rates $\epsilon \in [1, \beta]$ (tough part, not in this talk).
- We get the following picture [Lenzen et al., PODC 2009]:

<table>
<thead>
<tr>
<th>max rate $\beta$</th>
<th>$1+\epsilon$</th>
<th>$1+\Theta(\epsilon)$</th>
<th>$1+\sqrt{\epsilon}$</th>
<th>2</th>
<th>large</th>
</tr>
</thead>
<tbody>
<tr>
<td>local skew</td>
<td>$\infty$</td>
<td>$\Theta(\log D)$</td>
<td>$\Theta(\log_{1/\epsilon} D)$</td>
<td>$\Theta(\log_{1/\epsilon} D)$</td>
<td>$\Theta(\log_{1/\epsilon} D)$</td>
</tr>
</tbody>
</table>

We can have both smooth and accurate clocks! ... because too large clock rates will amplify the clock drift $\epsilon$.

- In practice, we usually have $1/\epsilon \approx 10^4 > D$. In other words, our initial intuition of a constant local skew was not entirely wrong! 😊
Clock Synchronization vs. Car Coordination

- In the future cars may travel at high speed despite a tiny safety distance, thanks to advanced sensors and communication.
Clock Synchronization vs. Car Coordination

- In the future cars may travel at high speed despite a tiny safety distance, thanks to advanced sensors and communication

- How fast & close can you drive?

- Answer possibly related to clock synchronization
  - clock drift ↔ cars cannot control speed perfectly
  - message jitter ↔ sensors or communication between cars not perfect
Example: Clock Synchronization?!?
...it's about TIME!

Roger Wattenhofer
One Big Difference Between Theory and Practice, Usually!

Physical Reality...

Worst Case Analysis!

Practice

Theory

$\Omega \left( \frac{W}{\sqrt{n \log n}} \right)$

$P \neq NP$
„Industry Standard“ FTSP in Practice

- As we have seen FTSP does have a local skew problem
- But it’s not all that bad...

- However, tests revealed another (severe!) problem:
- FTSP does not scale: Global skew grows exponentially with network size...
Experimental Results for Global Skew

FTSP

PulseSync

[Lenzen, Sommer, W, SenSys 2009]
Experimental Results for Global Skew

FTSP

PulseSync

[Lenzen, Sommer, W, SenSys 2009]
The PulseSync Protocol

1) Remove self-amplifying of synchronization error
2) Send fast synchronization pulses through the network
   - Speed-up the initialization phase
   - Faster adaptation to changes in temperature or network topology

\[
\text{Expected time} = D \cdot B / 2
\]

\[
\text{Expected time} = D \cdot t_{\text{pulse}}
\]

[Lenzen et al., SenSys 2009]
Experimental Results

- Synchronization error vs. hop distance

![Graph showing synchronization error vs. hop distance for FTSP and PulseSync.]
Everybody’s expectation, five years ago (solved)

Lower bound of $\log D / \log \log D$
[Fan & Lynch, PODC 2004]

All natural algorithms
[Locher et al., DISC 2006]

Blocking algorithm

Kappa algorithm
[Lenzen et al., FOCS 2008]

Dynamic Networks!
[Kuhn et al., SPAA 2009]

Tight lower bound
[Lenzen et al., PODC 2009]

Summary

FTSP
PulseSync
Thanks to my co-authors
Nicolas Burri
Christoph Lenzen
Thomas Locher
Philipp Sommer
Pascal von Rickenbach